PASSED

# GroupReport

*Akshayaswuroupikka Balasubramanian,yixuan xu*

*February 4, 2016*

## Assignment 1: Be careful with '=='

A pupil of a school is bad in arithmetic but good in programming. He writes a program to check if 1/3-1/4==1/12:

## 1.1

The result of the program is "Teacher lied".The computer calculates the value of 1/3 which is not a ture value. The bits allocated to the significand would provide roughly 16 decimal places of precision.

```
## [1] "Teacher lied"
```

## 1.2

The code is modified to give the correct result as follows:

```
## [1] "Teacher said true"
```

There is also another method in which the problem could be solved.Here we would do it by assigning a set significand to 16 bits.

```
## [1] "Teacher said true"
```

## Assignment 2: Derivative

A widely known way to compute the derivative of function f(x) in point x is to use

$$f'(x) = \frac{f(x+\epsilon) - f(x)}{\epsilon}$$

## 2.1

The function computing the derivative of function f(x)=x in this way is written below. Here we take $\epsilon = 10^{-15}$.

```
f<-function(x){
  e=10^-15
  a=x+e
  f_x=(a-x)/e
  print(f_x)
}
```

## 2.2

The derivative function at point x=100000 is computed.

```
## [1] 0
```

## 2.3

Here we get value 0, but the true value should be 1.From the function, when we calculate the (x+10^-15)-x, it is not equal to 10^-15, it becomes 0, this is because the significand has changed.

### Assignment 3: Variance

A known formula for estimating the variance is

$$Var(x) = \frac{1}{1-n}\left(\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right)$$

## 3.1

The function myvar() estimating the variance is given below:

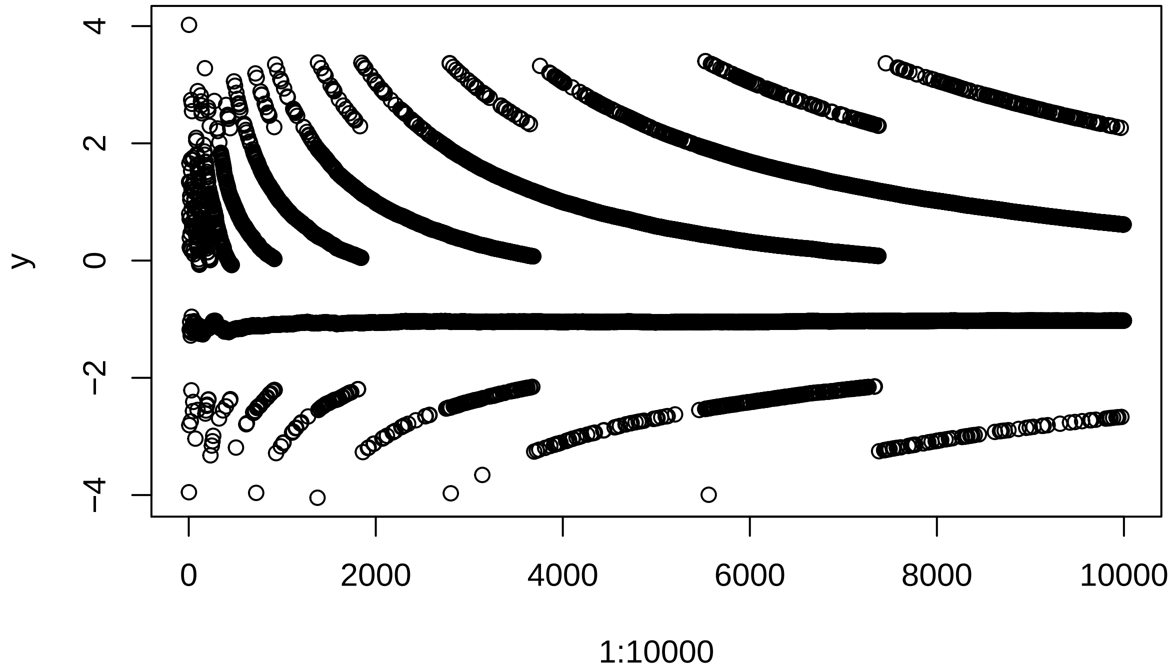
```
myvar <- function(x){
  n <- length(x)
  result <- (sum(x^2)-((sum(x)^2)/n))/(n-1)
  return(result)
}
```

## 3.2

A vector x=( x1…x10000) with 10000 random numbers, normally distributed with mean $10^8$ and variance 1 is generated.

## 3.3

The differences are round between (-4, 4). The true result should be 0.The error comes from the summation part. The precision of the summation is lost.The numbers are all of the same sign and have roughly the same magitude.In this case a pairwise "fan-in" method may yield good accuracy.And there is another error called cancellation, because the sum of square and square of sum has large magnitudes and small difference.The function does not go well for the above provided mean but ofcourse it works good for smaller mean values.



### Assignment 4: Linear Algebra

The Excel file “tecator.xls” contains the results of a study aimed to investigate whether a near- infrared absorbance spectrum and the levels of moisture and fat can be used to predict the protein content of samples of meat. For each meat sample the data consists of a 100 channel spectrum of absorbance records and the levels of moisture (water), fat and protein. The absorbance is -log10 of the transmittance measured by the spectrometer. The moisture, fat and protein are determined by analytic chemistry. The worksheet you need to use is “data”. It contains data from 215 samples of finely chopped meat. The aim is to fit a linear regression model that could predict protein content.

## 4.1

The data set is successfully imported in R.

## 4.2

A and b has been computed for the given data set.

## 4.3

The kind of result we are getting is:

Error in solve.default(a = A, b = b) : system is computationally singular: reciprocal condition number = 7.78804e-17

The reciprocal condition number is equal to 7.78804e-17 which could be consider as 0. Matrices are well-conditioned if the reciprocal condition number is near 1 and ill-conditioned if it is near zero.

## 4.4

```
## [1] 8.523517e+14
```

The conditional number of matrix A is found to be 8.523517e+14.When the conditiion number is very large it is said to be ill-conditioned.

## 4.5

```
##                    [,1]
## Channel1    -110.6420066
## Channel2    -221.1899921
## Channel3     378.0067049
## Channel4    -129.7038290
## Channel5     413.4180272
## Channel6     -79.7519946
## Channel7    -203.0060010
## Channel8      82.7949648
## Channel9    -132.3810763
## Channel10    255.8233113
## Channel11   -328.6085020
## Channel12   -304.1498067
## Channel13    624.1267225
## Channel14   -298.8990298
## Channel15     40.7433672
## Channel16   -257.5359421
## Channel17    169.2389191
## Channel18    296.6629389
## Channel19   -325.0660101
## Channel20     -3.0077434
## Channel21    554.5604351
## Channel22  -1366.0281946
## Channel23   1860.3564505
## Channel24  -1416.1321099
## Channel25    631.8396619
## Channel26   -112.0439549
## Channel27     17.0167193
## Channel28   -228.9306325
## Channel29    444.2724259
## Channel30   -597.3805369
## Channel31    438.1484415
## Channel32    315.0394088
## Channel33   -349.8143768
## Channel34   -285.9109757
## Channel35    418.5810505
## Channel36    -79.1068236
## Channel37   -305.9413388
## Channel38    284.2543473
## Channel39   -435.5657863
## Channel40    819.7486282
## Channel41   -885.0073314
## Channel42    324.5893411
## Channel43    524.5895670
## Channel44   -583.4418903
## Channel45   -140.1709745
## Channel46    577.2361744
## Channel47   -294.2684468
## Channel48    -68.0751275
## Channel49    -90.4922841
## Channel50    404.1439524
## Channel51   -698.9990527
## Channel52   1258.8833783
## Channel53  -1672.7308401
## Channel54   1486.2299172
## Channel55   -812.3613443
## Channel56    192.4954799
## Channel57    -32.9120466
## Channel58      7.3752546
## Channel59    -88.6907142
## Channel60    344.8769021
## Channel61   -454.3518607
## Channel62    447.6205296
## Channel63   -197.4186847
## Channel64    222.3370792
## Channel65   -399.2558318
## Channel66    364.8665574
## Channel67   -367.1614830
## Channel68    243.9220622
## Channel69    -76.2948345
## Channel70   -318.1916071
## Channel71    327.6653346
## Channel72   -178.5231528
## Channel73    119.1856499
## Channel74    445.1150045
## Channel75    -20.0127319
## Channel76   -642.7509961
## Channel77    369.4809508
## Channel78    -74.9011366
## Channel79    -23.4854322
## Channel80   -676.8615334
## Channel81   1013.4538029
## Channel82   -889.7623189
## Channel83    403.0065633
## Channel84    424.0848303
## Channel85   -801.0956155
## Channel86    655.0134202
## Channel87    659.1829785
## Channel88  -2150.8325647
## Channel89   1671.8088853
## Channel90    298.6977068
## Channel91   -332.1727762
## Channel92   -487.3689759
## Channel93    278.6277368
## Channel94    201.6627353
## Channel95   -609.5081456
## Channel96    565.2851789
## Channel97   -133.3407595
## Channel98   -368.0087250
## Channel99    238.2015968
## Channel100    24.6418188
## Fat           -1.6666403
## Moisture      -0.9341099
```

```
## [1] 490471520662
```

For the original data the fat,moisture have large magnitude and the channels have smaller magnitude.After the scaling data set, the columns of the matrix have the same scale, which avoids artificial ill-conditioning.

# Appendix - R-code

```
## ----echo=FALSE---------------------------------------------------------
x1<-1/3
x2<-1/4;
if (x1-x2==1/12){
print("Teacher said true")
} else{
print("Teacher lied")}

## ---- echo=FALSE--------------------------------------------------------
x1<-1/3
x2<-1/4
if(all.equal(x1-x2,1/12)){
   print("Teacher said true")
} else {
   print("Teacher lied")
}

## ----echo=FALSE---------------------------------------------------------
x1<-1/3
x2<-1/4
if (round(x1-x2, digits = 16) == round(1/12, digits = 16)){
  print("Teacher said true")
} else{
  print("Teacher lied")}

## -------------------------------------------------------------------------
f<-function(x){
  e=10^-15
  a=x+e
  f_x=(a-x)/e
```

```r
  print(f_x)
}

## ----echo=FALSE----------------------------------------------------------
f(100000)

## ------------------------------------------------------------------------
myvar <- function(x){
  n <- length(x)
  result <- (sum(x^2)-((sum(x)^2)/n))/(n-1)
  return(result)
}

## ----echo=FALSE----------------------------------------------------------
x <- rnorm(10000,mean = 10^8, 1)

## ----echo=FALSE----------------------------------------------------------
y <- NULL
for(i in 1 : 10000){
a <- myvar(x[1:i])
b <- var(x[1:i])
y[i] <- a - b
}
plot(x= 1: 10000,y=y)

## ----echo=FALSE----------------------------------------------------------
data <- read.csv("tecator.csv")

## ----echo=FALSE----------------------------------------------------------
X <- as.matrix(data[,c(2:102,104)])
Y <- as.matrix(data[,103])
X_1 <- as.matrix(cbind(1,X))
A <- t(X_1)%*%X_1
b <- t(X_1)%*%Y

## ----eval=FALSE,echo=FALSE-----------------------------------------------
## beta <- solve(A, b)

## ----echo=FALSE----------------------------------------------------------
kappa(A)

## ----echo=FALSE----------------------------------------------------------
X1 <- scale(X, center = TRUE, scale = TRUE)
Y1 <- scale(Y, center=TRUE, scale=TRUE)
A1 <-t(X1)%*%X1
b1 <- t(X1)%*%Y1
beta1 <- solve(a = A1, b = b1)
beta1
kappa(A1)

## ----code=readLines(knitr::purl("GroupReport.Rmd", documentation = 1)), eval = FALSE----
## NA
```